

Remaining Useful Life Estimation of Batteries using Dirichlet Process with Variational Bayes Inference

Pajovic, Milutin; Orlik, Philip V.; Wada, Toshihiro

TR2018-152 November 13, 2018

Abstract

Rechargeable batteries supply numerous devices with electric power and are critical part in a variety of applications. While estimation of battery's state of charge (SoC), state of health (SoH) and state of power (SoP) have been in research focus in the past years, prediction of battery degradation has recently started to gain interest. An accurate prediction of the remaining number of charge and discharge cycles a battery can undergo before it can no longer hold charge and is declared dead, is directly related to making timely decision as to when a battery should be replaced so that power interruption of the system it supplies power to is avoided. A methodology for inferring probability distribution of the remaining number of charge-discharge cycles of a battery, based on training dataset containing measured discharge voltage waveforms of one or more batteries of similar type, is presented in this paper. The methodology strongly draws on modeling discharge voltage waveforms using Dirichlet Process Mixture Model framework and performs approximate inference using variational Bayes approach. The experimental results corroborate that the proposed method is able to provide useful predictions of the remaining useful life of a battery in early stages of its life.

Annual Conference of the IEEE Industrial Electronics Society (IECON)

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Remaining Useful Life Estimation of Batteries using Dirichlet Process with Variational Bayes Inference

Milutin Pajovic

Mitsubishi Electric Research Labs
Cambridge, MA, USA
pajovic@merl.com

Philip Orlik

Mitsubishi Electric Research Labs
Cambridge, MA, USA
porlik@merl.com

Toshihiro Wada

Mitsubishi Electric Corporation
Amagasaki City, Japan
Wada.Toshihiro@bx.MitsubishiElectric.co.jp

Abstract—Rechargeable batteries supply numerous devices with electric power and are critical part in a variety of applications. While estimation of battery’s state of charge (SoC), state of health (SoH) and state of power (SoP) have been in research focus in the past years, prediction of battery degradation has recently started to gain interest. An accurate prediction of the remaining number of charge and discharge cycles a battery can undergo before it can no longer hold charge and is declared dead, is directly related to making timely decision as to when a battery should be replaced so that power interruption of the system it supplies power to is avoided. A methodology for inferring probability distribution of the remaining number of charge-discharge cycles of a battery, based on training dataset containing measured discharge voltage waveforms of one or more batteries of similar type, is presented in this paper. The methodology strongly draws on modeling discharge voltage waveforms using Dirichlet Process Mixture Model framework and performs approximate inference using variational Bayes’ approach. The experimental results corroborate that the proposed method is able to provide useful predictions of the remaining useful life of a battery in early stages of its life.

Index Terms—rechargeable battery, remaining useful life, charge-discharge cycle, Dirichlet process mixture model, variational Bayes’, probabilistic inference

I. INTRODUCTION

Rechargeable batteries nowadays supply power to a variety of systems such as electric vehicles, consumer electronic devices, uninterrupted power supply (UPS) systems, etc. They also support photo-voltaic systems and smart power grids. Developing methods for estimating battery state of charge (SoC), state of power (SoP) and state of health (SoH) has been in persistent research focus in the past years [1], [2]. Predicting battery degradation over time is of utmost importance in a number of applications and has recently started to gain research interest. More specifically, the problem is to accurately predict how long a battery can supply its load, usually expressed with the number of remaining charge and discharge cycles it can undergo. This ensures that the battery is replaced on time, thus avoiding (1) power disruptions due to unexpected battery failure and (2) discarding the battery before its useful life ends. A related problem, also relevant in a number of applications, is concerned with predicting the remaining time to discharge of a battery [12], preferably early in the discharge cycle.

A concise review of methods for battery remaining useful life estimation is given in [3]. In the domain of our interest, the

prognostics of battery’s state of health has been approached from Bayesian perspective in [4]. Particle filtering is used in a variety of contexts to estimate the remaining useful life of a battery. As such, [5] tracks battery degradation using a regular particle filter, [6] employs spherical cubature particle filter, [7] designs risk sensitive particle filter, while [8] exploits unscented particle filter. A state-of-health regeneration phenomenon is modeled and predicted using suitably designed particle filter [9]. On-line fault diagnosis and failure prognosis is also approached with particle filter in [10]. In a similar vein, diagnosis and prognosis is done using Lebesgue sampling in [11].

In this paper, we propose an algorithm for inferring remaining useful life (RUL) of a battery using Dirichlet Process Mixture Model (DPMM) with variational Bayes (VB) inference. More specifically, we cluster feature vectors representing discharge voltage waveforms of one or more batteries measured during their lifetimes. The obtained clusters indicate types of possible aging stages a battery goes through during its lifetime. In the operational stage, upon representing a measured discharge voltage waveform of a battery of similar type with a feature vector, the discharge cycle is probabilistically associated with one of the clusters, i.e., aging stage. Consequently, the remaining number of charge-discharge cycles the battery can further undergo is predicted based on the estimated aging stage.

The works most relevant to this paper are [12] and [13]. While we adopt the same empirical model for discharge voltage waveform and generative model for feature vectors as [12], we address a completely different problem. Namely, we infer remaining useful life (RUL) of a battery at the end of some discharge cycle, while [12] estimates remaining time to full discharge of a battery at some time instant early in the discharge cycle. In addition, [12] performs the DPMM inference using Gibbs’ sampling, while we perform inference using variational Bayes’. A DPMM-based algorithm for estimating remaining number of charge-discharge cycles of a battery is also proposed in [13]. In comparison to [13], we use different empirical model for discharge voltage and assume different generative model for feature vectors. Furthermore, [13] performs DPMM inference using Gibbs’ sampling, while the models we adopt admit faster inference based on variational Bayes’.

The paper is organized as follows. Section II describes feature vectors used to represent battery aging. Section III describes the approach used for inferring remaining useful life of a battery. Section IV presents our method based on DPMM modeling and variational Bayes' inference. Section V experimentally validates the proposed method. Section VI concludes the paper.

II. FEATURE VECTORS FOR BATTERY AGING

We use experimental battery dataset collected and made publicly available by the NASA Ames Research Center of Excellence [18]. The dataset contains measured voltage and current waveforms taken during charge and discharge cycles of various batteries undergoing accelerated aging, starting from nominal capacity of 2 Ah and ending when the capacity falls below 1.6 Ah, at which point a battery is declared dead. In addition to voltage and current measurements, the dataset also contains battery capacity recording at the end of each discharge cycle, evaluated using Coulomb counting, as well as the battery impedance measurements. We use discharge voltage measurements as proxy to battery aging.

A voltage waveform, recorded during a single discharge cycle, contains few hundreds of samples. While the method described here can, in general, be applied directly to the recorded discharge voltage, a computationally less demanding and analytically more tractable method results from representing each discharge voltage waveform with parameters of an empirical model for the discharge voltage waveform. We utilize empirical model from [12], which models the discharge voltage $V(t)$ as

$$V(t) = E_0 - a_1 e^{-a_2/t} - a_3 e^{a_4 t} + a_5 t, \quad (1)$$

where t is the time such that $t \rightarrow 0$ corresponds to the beginning of the discharge cycle and E_0 is the voltage of a fully charged battery, where $E_0 = 4.2$ V for dataset [18]. The model parameters $a_j, j = 1, \dots, 5$, are all real-valued and constitute a feature vector $\mathbf{a} = [a_1 \ a_2 \ a_3 \ a_4 \ a_5] \in \mathbb{R}^5$. Therefore, each discharge cycle i in the dataset is represented with feature vector \mathbf{a}_i , also referred to as the data point, by fitting $V(t)$, measured during that discharge cycle, with the empirical model (1). Consequently, only five model parameters per discharge cycle are stored instead of the whole waveform $V(t)$. More importantly, model training and estimation of the battery's remaining useful life operates over five dimensional feature vectors \mathbf{a}_i instead of much longer discharge voltage waveforms.

As a side note, we emphasize that direct measurements of other quantities that depend on battery aging, or the parameters of the corresponding empirical models can be used for estimation of battery's remaining useful life. The notion of remaining useful life of a battery and outline of an approach for its probabilistic inference are described in the following part.

III. INFERENCE OF REMAINING USEFUL LIFE

The remaining useful life (RUL) of a battery at some time instant is a random variable that represents the number of

charge-discharge cycles the battery can undergo from that time instant until it can no longer hold the charge and is declared dead. In the problem setup considered here, the battery is being charged and discharged according to some predefined template, and the aim is to infer the RUL at the end of a discharge cycle, based on the measured voltage during that discharge cycle. The charge-discharge template specifies how a battery is charged and discharged, for example, a battery in the NASA's dataset is charged with current 1.5 A until its voltage reaches 4.2 V, which is then kept constant until the current drops below 20 mA, while the discharge is performed with constant discharge current of 2 A until the voltage drops below 2.5 V.

The RUL of a battery in operational/online stage is inferred using training data that contains feature vectors representing measured discharge voltage waveforms during discharge cycles of one or more batteries of similar type. More specifically, the training data \mathcal{D} contains N pairs

$$\mathcal{D} = \{(\mathbf{a}_1, k_1), (\mathbf{a}_2, k_2), \dots, (\mathbf{a}_N, k_N)\}, \quad (2)$$

where \mathbf{a}_n is the feature vector computed from measured voltage after some discharge cycle, while k_n is the number of remaining charge-discharge cycles left after that discharge cycle. In the case the training data \mathcal{D} is populated with discharge voltage measurements from only one battery over its useful life, N is the overall number of discharge cycles that battery underwent during its lifetime, n is the discharge cycle index such that $n = 1$ corresponds to the first discharge cycle, while $k_n = N - n$. In general, \mathcal{D} may contain measurements from more than one battery, in which case N is the total number of measured discharge voltage waveforms, while n indexes data points.

Given training data \mathcal{D} , the RUL of a battery at the end of some discharge cycle in the operational/online stage is inferred by extracting k_n 's from \mathcal{D} whose corresponding \mathbf{a}_n 's are close in some sense to feature vector \mathbf{a}' representing measured discharge voltage during that discharge cycle. This is robustly done by clustering data from \mathcal{D} in the training stage, and classifying the feature vector \mathbf{a}' into one of those clusters in the operational stage. Formally, assume \mathcal{D} is clustered into L clusters and let \mathcal{K}_l denote the set of remaining number of charge-discharge cycles corresponding to training data points that constitute cluster l . More specifically, if some \mathbf{a}_i belongs to cluster l , then the corresponding $k_i \in \mathcal{K}_l$.

Since a cluster l contains data points whose corresponding numbers of remaining charge-discharge cycles are different (although relatively close), the remaining useful life of points from cluster l is represented with a probability distribution

$$p_{x|l}(x|l) \propto \sum_{k \in \mathcal{K}_l} g(x, k), \quad (3)$$

where $g(x, k)$ is a kernel centered at $k \in \mathcal{K}_l$, and x is the number of remaining charge-discharge cycles. Without loss of generality, we use Gaussian kernel

$$g(x, k) = e^{-\frac{(x-k)^2}{2\sigma^2}}, \quad (4)$$

where σ^2 is an appropriately selected kernel width. Other kernel functions are possible, for example, a discrete delta function $\delta(x, k)$ (equal to 1 if $x = k$ and zero otherwise), yields normalized histogram over k 's from \mathcal{K}_l .

In the online/operational stage, measured discharge voltage of a battery is fitted with empirical model (1), resulting in feature vector \mathbf{a}' . The feature vector \mathbf{a}' is then softly classified into clusters l , yielding probability $p_{\mathbf{a}}(\mathbf{a}' \in l)$ that \mathbf{a}' comes from cluster l , $l = 1, \dots, L$. Finally, the remaining useful life of the battery is evaluated as the weighted combination of $p_{x|l}(x|l)$, $l = 1, \dots, L$, where the weights are cluster probabilities $p(\mathbf{a}' \in l)$,

$$p_{\text{RUL}}(x|\mathbf{a}') = \sum_{l=1}^L p_{\mathbf{a}}(\mathbf{a}' \in l) p_{x|l}(x|l) \quad (5)$$

Given the described framework, it remains to specify the method for clustering training data \mathcal{D} . In general, the RUL framework places on constraints on the clustering method that can be used. However, for the reasons that will become clear later we employ a powerful Dirichlet Process Mixture Model (DPMM). The following section provides details on the DPMM clustering method.

IV. DATA MODEL AND INFERENCE

This section provides details on generative model assumed for feature vectors \mathbf{a} 's, describes clustering of the training data \mathcal{D} , and outlines a method for computing class probabilities $p(\mathbf{a}' \in l)$ of feature points acquired in the online stage.

A. DPMM Generative Model for Data Points

We assume data points \mathbf{a}_i , $i = 1, \dots, N$ are generated according to the Dirichlet Process Mixture Model (DPMM) [15]–[17]. The Dirichlet Process (DP) is a generative model specified with concentration parameter α and base distribution H , often denoted as $\text{DP}(\alpha, H)$. One may think of a DP as a probability distribution over probability distributions such that a sample from the DP is an infinite dimensional discrete probability distribution. The support of such discrete probability distribution is generated according to the base distribution H , while the values of probability masses are controlled by the concentration parameter α . Each support element in the DP is a cluster center of the DPMM, while its probability mass is the corresponding cluster probability.

More specifically, a data point \mathbf{a}_i is generated by sampling a cluster c_i according to cluster probabilities π_l , $l = 1, \dots, L$, and then sampling from Gaussian distribution, parameterized with the cluster center (which in turn is generated from H). This results in data point \mathbf{a}_i being associated with cluster c_i and generated as a sample from, in our case, Gaussian distribution with mean vector $\mathbf{\Lambda}_{c_i}$ and covariance matrix $\mathbf{\Sigma}_{c_i}$,

$$p(\mathbf{a}_i|c_i, \mathbf{\Lambda}_{c_i}, \mathbf{\Sigma}_{c_i}) = \mathcal{N}(\mathbf{a}_i; \mathbf{\Lambda}_{c_i}, \mathbf{\Sigma}_{c_i}) \quad (6)$$

We emphasize that $(\mathbf{\Lambda}_{c_i}, \mathbf{\Sigma}_{c_i})$ are associated with the cluster c_i . The generative model for mean vector and covariance matrix associated with each cluster l is chosen such that the prior $p(\mathbf{\Lambda}_{c_i}, \mathbf{\Sigma}_{c_i})$ is conjugate to model $p(\mathbf{a}_i|c_i, \mathbf{\Lambda}_{c_i}, \mathbf{\Sigma}_{c_i})$ so

as to yield tractable approximate inference. Therefore, the covariance matrix $\mathbf{\Sigma}_{c_i}$ is given by [12]

$$\mathbf{\Sigma}_{c_i} = \text{diag}\{s_{c_i,1}^{-1}, \dots, s_{c_i,5}^{-1}\} \quad (7)$$

where $s_{c_i,k} \sim \text{Gamma}(\beta, \gamma)$, $k = 1, \dots, 5$, are independent samples from Gamma distribution with hyperparameters β and γ . The mean vector $\mathbf{\Lambda}_{c_i}$ is generated according to

$$p(\mathbf{\Lambda}_{c_i}|\mathbf{\Sigma}_{c_i}) = \mathcal{N}(\mathbf{\Lambda}_{c_i}; \mathbf{a}_0, h\mathbf{\Sigma}_{c_i}), \quad (8)$$

where \mathbf{a}_0 is same across all clusters and h is a hyperparameter. This generative model for cluster centers $(\mathbf{\Lambda}_{c_i}, \mathbf{\Sigma}_{c_i})$ embodies base distribution H . While the selection of H may seem arbitrary and not much related to our original problem, we point out that, in general, a generative model with hierarchical structure for observed data is robust to selection of hyperparameters.

The number of clusters in the DPMM is not fixed in advance and is inferred from data. The DPMM generates clusters according to a stick-breaking process, wherein an imagined unit-length stick is being broken into a growing number of segments such that each segment represents one cluster with probability equal to the length of the corresponding segment. Specifically, the probability π_l of cluster l is given by [15]

$$\pi_l = v_l \prod_{j=1}^{l-1} (1 - v_j), \quad (9)$$

where $v_l \sim \text{Beta}(1, \alpha)$. Thus, at the beginning of the stick-breaking process, we are given a unit length stick and break a segment from it of length $v_1 \sim \text{Beta}(1, \alpha)$, which gives the probability of the first cluster. The remaining segment of length $1 - v_1$ is then subject to stick breaking in the second round, so that the length of the segment obtained after breaking a segment of length $v_2 \sim \text{Beta}(1, \alpha)$ is $\pi_2 = v_2(1 - v_1)$ and is equal to the second cluster probability. The process continues in the same manner and results in infinitely many stick segments, the length of each is equal to one cluster probability.

Finally, we emphasize that although the DPMM has potential to generate infinitely many clusters, we observe finite number of data points N and consequently can see only a finite number of clusters. In fact, one may think that a newly observed data point \mathbf{a} comes from one of the existing clusters, or from some previously unseen cluster. The rate at which new clusters are created is directly related to cluster probabilities π_l and both are controlled by the concentration parameter α . Intuitively, the larger probabilities of existing clusters, the smaller chance a newly acquired data point comes from unseen cluster. We model the concentration parameter as $\alpha \sim \text{Gamma}(s_1, s_2)$ to fully adhere to hierarchical generative model so that the inference is less sensitive to the choice of hyperparameters, in this case, s_1 and s_2 than it would be in case α was itself a hyperparameter.

B. DPMM Approximate Inference

Given training dataset \mathcal{D} , the goal of the training stage is to infer posterior distribution of cluster centers $\mathbf{\Lambda}_l$ and $\mathbf{\Sigma}_l$, cluster probabilities implicitly given by v_l , cluster indices c_i of training data points \mathbf{a}_i , and concentration parameter α . Formally, the aim is to find

$$p \triangleq p(\{c_i\}_{i=1}^N, \{\mathbf{\Lambda}_l, \mathbf{\Sigma}_l\}_{l=1}^L, \{v_l\}_{l=1}^L, \alpha | \mathcal{D}), \quad (10)$$

where L is the implicit number of clusters observed/seen in the data. Since solving (10) using Bayes' rule is intractable, we need to resort to approximations. Two main approaches for approximate inference are Markov Chain Monte Carlo (MCMC) and variational Bayes (VB) [14]. The generative model described in the previous part employs conjugate priors and is thus suitable for Gibbs' sampling, as done in [12], because all posterior conditionals that the Gibbs' sampler samples from are standard distributions. Nevertheless, we utilize the mean field approximation (MFA) algorithm, which is the most popular VB method because the MFA is faster than the Gibbs' sampling. In addition, the MFA yields closed form expressions for iterative parameter updates due to conjugacy in the data generative model.

The MFA approximates the true posterior p with a distribution that factorizes over all unknown variables,

$$p \approx q(\{c_i\}_{i=1}^N, \{\mathbf{\Lambda}_l, \mathbf{\Sigma}_l\}_{l=1}^L, \{v_l\}_{l=1}^L, \alpha) = \prod_{i=1}^N q_{c_i}(c_i) \prod_{l=1}^L q_{\mathbf{\Lambda}_l}(\mathbf{\Lambda}_l) \prod_{l=1}^L q_{\mathbf{\Sigma}_l}(\mathbf{\Sigma}_l) \prod_{l=1}^L q_{v_l}(v_l) q_{\alpha}(\alpha) \quad (11)$$

The approximating distribution is found by minimizing the Kullback-Liebler (KL) divergence between the two distributions, $\text{KL}(q||p)$ [14]. Using the calculus of variations and some algebraic manipulations yield expressions for computing each factor in q . For example, the resulting approximating distribution for c_k is given by

$$\log q_{c_k}(c_k) \propto \mathbb{E} [\log p(\mathcal{D}, \{c_i\}_{i=1}^N, \{\mathbf{\Lambda}_l, \mathbf{\Sigma}_l\}_{l=1}^L, \{v_l\}_{l=1}^L, \alpha)], \quad (12)$$

where the expectation is taken with respect to $\prod_{i \neq k} q_{c_i} \prod_l q_{\mathbf{\Lambda}_l} \prod_l q_{\mathbf{\Sigma}_l} \prod_l q_{v_l} q_{\alpha}$, i.e., with respect to the product of all factors in q excluding q_{c_k} . The update expressions for other factors in q are given with analogous expressions.

The distribution under the expectation operator in (12) is joint likelihood of the observed data \mathcal{D} and unknown parameters, and can be expressed as the sum of logarithm terms by exploiting the conditional independences embedded in the generative model. Once that is done, each factor term in q is computed by taking the expectation of the resulting expression with respect to the product of all other factor terms. We omit the derivation details as they are tedious and do not lend any particular insights. In the following, we provide the resulting expressions for the factors in the approximating distribution q .

The factor distributions $q_{\mathbf{\Lambda}_l}$ and $q_{\mathbf{\Sigma}_l}$ that approximate posteriors of the mean vector $\mathbf{\Lambda}_l$ and covariance matrix $\mathbf{\Sigma}_l$

factorize, respectively, over their entries $\Lambda_{l,k}$ and $s_{l,k}$, where $k = 1, \dots, 5$, such that

$$\Lambda_{l,k} \sim \mathcal{N}(\mu_{l,k}, \sigma_{l,k}^2) \text{ and } s_{l,k} \sim \text{Gamma}(\beta_{l,k}, \gamma_{l,k}) \quad (13)$$

Furthermore, the approximate posteriors q_{v_l} and q_{α} are given by

$$v_l \sim \text{Beta}(\delta_l, \alpha_l) \text{ and } \alpha \sim \text{Gamma}(s_1, s_2) \quad (14)$$

The approximation for the posterior probability $p_{i,l} \triangleq q_{c_i}(l)$ that the data point \mathbf{a}_i belongs to cluster l , is evaluated as

$$\begin{aligned} \log p_{i,l} \propto & \frac{1}{2} \sum_{k=1}^5 \mathbb{E}[\log s_{l,k}] - \frac{1}{2} \mathbb{E}[s_{l,k}] \mathbb{E}[(a_{i,k} - \Lambda_{l,k})^2] \\ & + \mathbb{E}[\log v_l] + \sum_{n=1}^{l-1} \mathbb{E}[\log(1 - v_l)], \end{aligned} \quad (15)$$

where $a_{i,k}$ is the k th entry in \mathbf{a}_i . Above, given that $s_{l,k}$ and v_l are, respectively, Gamma and beta distributed, $\mathbb{E}[s_{l,k}] = \beta_{l,k}/\gamma_{l,k}$, $\mathbb{E}[\log s_{l,k}] = \psi(\beta_{l,k}) - \log(\gamma_{l,k})$ and $\mathbb{E}[\log v_l] = \psi(\delta_l) - \psi(\delta_l + \alpha_l)$, where $\psi(x)$ is Digamma function.

Thus, all factors from the approximating distribution q follow standard distributions. The parameters of those distributions are also obtained from (12) and alike expressions. Omitting the derivation details, these parameters are iteratively updated as

$$\mu_{l,k} = \frac{h \sum_{i=1}^N a_{i,k} p_{i,l} + a_{0,k}}{h \sum_{i=1}^N p_{i,l} + 1} \quad (16)$$

$$\sigma_{l,k}^2 = \frac{h \gamma_{l,k}}{\beta_{l,k} (h \sum_{i=1}^N p_{i,l} + 1)} \quad (17)$$

$$\beta_{l,k} = \beta + 0.5 \left(\sum_{i=1}^N p_{i,l} + 1 \right) \quad (18)$$

$$\begin{aligned} \gamma_{l,k} = & \gamma + \frac{1}{2h} (a_{0,k}^2 - 2a_{0,k}\mu_{l,k} + \mu_{l,k}^2 + \sigma_{l,k}^2) \\ & + \frac{1}{2} \sum_{i=1}^N p_{i,l} (a_{i,k}^2 - a_{i,k}\mu_{l,k} + \mu_{l,k}^2 + \sigma_{l,k}^2) \end{aligned} \quad (19)$$

$$\delta_l = 1 + \sum_{i=1}^N p_{i,l} \quad (20)$$

$$\alpha_l = \frac{w_1}{w_2} + N - \sum_{i=1}^N \sum_{k=1}^l p_{i,k} \quad (21)$$

$$w_1 = s_1 + L - 1 \quad (22)$$

$$w_2 = s_2 - \sum_{k=1}^{L-1} (\psi(\alpha_k) - \psi(\delta_k + \alpha_k)), \quad (23)$$

where $\psi(x)$ is the Digamma function.

The iterative routine is initialized with uniform $p_{i,l}$ for each data point \mathbf{a}_i , while $w_1 = s_1$, $w_2 = s_2$ and the maximum number of clusters to look for in the dataset is set to some L . The hyperparameters s_1 , s_2 , β and γ are set to some small values so that the corresponding distributions are non-informative, while h is set so that the distribution in (8) is wide enough. Then, α_l , w_1 , w_2 , $\beta_{l,k}$, $\mu_{l,k}$, $\sigma_{l,k}^2$, $\gamma_{l,k}$ and $p_{i,l}$, where

$l = 1, \dots, L$, $k = 1, \dots, 5$ and $i = 1, \dots, N$ are iteratively updated according to (16)–(23). The iterative updates are run predefined number of iterations or until convergence is established. We note that some other ways of initializing the iterative routine and ordering in updating the factor parameters are also possible, however we believe the presented one is most suitable. Overall, the iterative procedure results in approximate posterior distributions of unknown parameters, conditioned on data \mathcal{D} . The next problem addressed in the following part is to compute cluster probabilities of a feature vector \mathbf{a}' , acquired in the operational stage, based on clustering results of data \mathcal{D} .

C. Cluster Probabilities of Test Data Points

In the testing/operational stage, measured voltage during some discharge cycle is fitted using the empirical model (1) and represented with feature vector \mathbf{a}' . Classifying \mathbf{a}' into clusters discovered in the training stage consists of computing cluster probability distribution $p_{\mathbf{a}}(\mathbf{a}' \in l)$, $l = 1, \dots, L$. The cluster probability is computed as

$$p_{\mathbf{a}}(\mathbf{a}' \in l) = \int p(\mathbf{a}|l, \Lambda_l, \Sigma_l) q_{\Lambda_l}(\Lambda_l) q_{\Sigma_l}(\Sigma_l) d\Lambda_l d\Sigma_l. \quad (24)$$

Since $q_{\Lambda_l}(\Lambda_l)$ and $q_{\Sigma_l}(\Sigma_l)$ factorize into products of constituent factors, the cluster probability is after a few steps of algebraic manipulations further given by

$$\begin{aligned} p_{\mathbf{a}}(\mathbf{a}' \in l) &= \prod_{k=1}^5 \mathbb{E}_{\Lambda_{l,k}, \Sigma_{l,k}} [\mathcal{N}(a'_k; \Lambda_{l,k}, \Sigma_{l,k})] \\ &= \prod_{k=1}^5 \mathbb{E}_{\Lambda_{l,k}, s_{l,k}} \left[\frac{\sqrt{s_{l,k}}}{2\pi} e^{-\frac{s_{l,k}}{2} (a'_k - \Lambda_{l,k})^2} \right] \end{aligned} \quad (25)$$

where $\Sigma_{l,k} = s_{l,k}^{-1}$ and a'_k is the k th entry in \mathbf{a}' .

The expectation in (25) is computed using the law of iterated expectations, where the expectation with respect to $s_{l,k}$ is computed first and expressed in terms of Gamma function of first kind $\Gamma(\cdot)$. Skipping the algebraic manipulations, this results in

$$\begin{aligned} p_{\mathbf{a}}(\mathbf{a}' \in l) &= \frac{1}{(2\pi)^{5/2}} \prod_{k=1}^5 \frac{\gamma_{l,k}^{\beta_{l,k}} \Gamma(\beta_{l,k} + 0.5)}{\Gamma(\beta_{l,k})} \times \\ &\mathbb{E}_{\Lambda_{l,k}} \left[\frac{1}{(\gamma_{l,k} + 0.5(a'_k - \Lambda_{l,k})^2)^{\beta_{l,k} + 0.5}} \right] \end{aligned} \quad (26)$$

Since the expectation in (26) is not given in a closed form, it is computed numerically by sampling $\Lambda_{l,k} \sim \mathcal{N}(\mu_{l,k}, \sigma_{l,k}^2)$.

This concludes the computation of cluster probability distribution $p_{\mathbf{a}}(\mathbf{a}' \in l)$. Substituting the resulting distribution into (5) yields probability distribution $p_{\text{RUL}}(x|\mathbf{a}')$ of the remaining useful life of a battery at the end of the discharge cycle corresponding to feature vector \mathbf{a}' .

V. EXPERIMENTAL RESULTS

The experimental testing of the described methodology is performed using battery data from NASA Ames Research Center of Excellence [18]. The measured discharge voltage waveforms of two selected batteries are represented with

parameters (feature vectors) that fit the specified empirical model (1). The feature vectors \mathbf{a} of one battery, labeled as battery B6 in the NASA's dataset, are part of the training data \mathcal{D} , while the feature vectors corresponding to the other battery, labeled as B5 in the NASA's dataset, are used to testing.

We start by clustering data points from \mathcal{D} using the DPMM generative model and the approximate inference, as detailed in Section IV. For that purpose, we set $h = 5$ so that the distribution that generates feature vectors \mathbf{a} is wide enough. The mean feature vector \mathbf{a}_0 is set to the mean of all feature vectors from \mathcal{D} . The hyper-parameters of Gamma distributions in the DPMM model are chosen so that the distributions are non-informative, $s_1 = \gamma = 10^{-7}$ and $s_2 = \beta = 1 + 10^{-7}$. One of the outputs from the DPMM inference routine are cluster probabilities $p_{i,l}$, representing the probability that a feature vector \mathbf{a}_i is generated from cluster l . They are shown in Fig. 1 such that its vertical slice is a resulting cluster probability distribution of the corresponding feature vector. We note that cluster indices have no particular meaning and some of them contain no data. In fact, out of $L = 20$ clusters that the inference procedure is initialized with, the result indicates the existence of only 10 clusters. Notably, most feature vectors are associated with high probability (close to 1) with only one cluster.

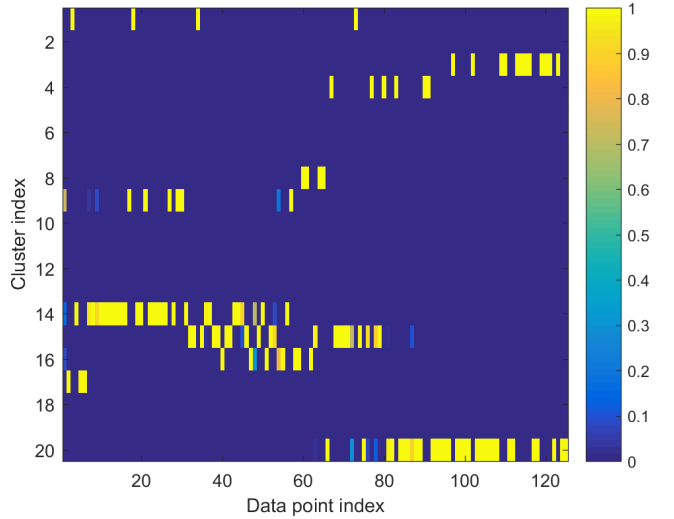


Fig. 1. DPMM data clustering probabilities of the training data (battery B6).

Each feature vector in \mathcal{D} is paired with the number of charge-discharge cycles that remain from the cycle that feature vector corresponds to until the end of the battery's useful life. The number of remaining charge-discharge cycles associated with feature vectors from the same cluster versus cluster index is shown in Fig. 2. As expected, most clusters cover a range of the number of remaining number of charge-discharge cycles. The exception is cluster 1 which contains 4 data points from \mathcal{D} whose corresponding remaining number of cycles cover relatively wide range, indicating that this cluster effectively collects outliers and is thus not useful for RUL estimation.

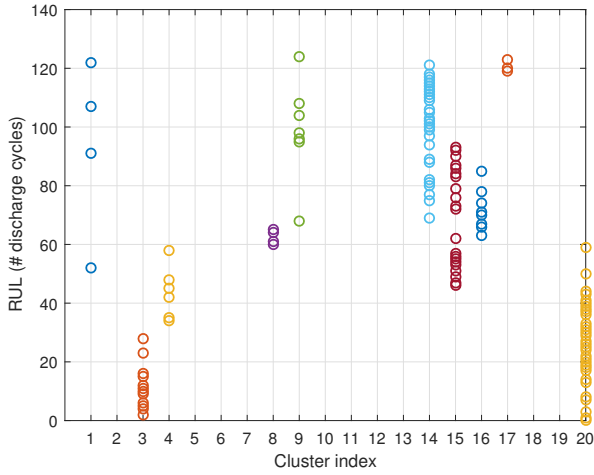


Fig. 2. Remaining number of charge-discharge cycles versus cluster index.

Each feature vector from the test data is softly classified into clusters identified in the training stage, as detailed in Section IV-C. The resulting clustering probability distributions are shown in Fig. 3. As can be observed, most of the test data points are, with probability close to one, associated with one cluster. Also, a fairly large number of test data points get clustered in a particular cluster (the one with index 20).

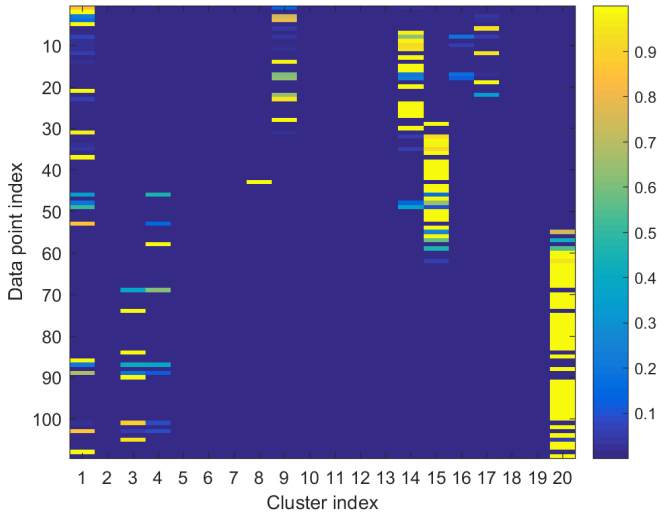


Fig. 3. Clustering of feature vectors from the test data (battery B5).

The RUL probability distribution is inferred at the end of each charge-discharge cycle of the testing data using (5) and the results shown in Fig. 3. We use $\sigma^2 = 4$ for the parameter in the Gaussian kernel (4) such that if a certain feature vector is associated with the remaining number of charge-discharge cycles k , it can also be associated with the remaining number of cycles $k - 1$ and $k + 1$, with relatively high probability. The RUL distribution at the end of the 10th charge-discharge cycle of the tested battery B5 is shown in Fig 4, where

the true number of remaining charge-discharge cycles is 99. Similarly, the RUL distribution at the end of the 80th cycle is shown in Fig. 5, where the true number of remaining cycles is 29. We observe that the RUL distributions in Figs. 4 and 5 are concentrated around the corresponding true numbers of remaining cycles and fairly accurately predict the remaining useful life of the tested battery at the tested time instants.

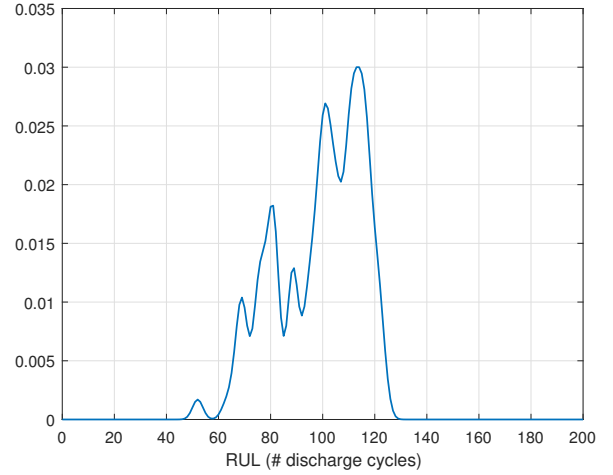


Fig. 4. Predicted probability distribution of the RUL at the end of 10th charge-discharge cycle. The true number of remaining cycles is 99.

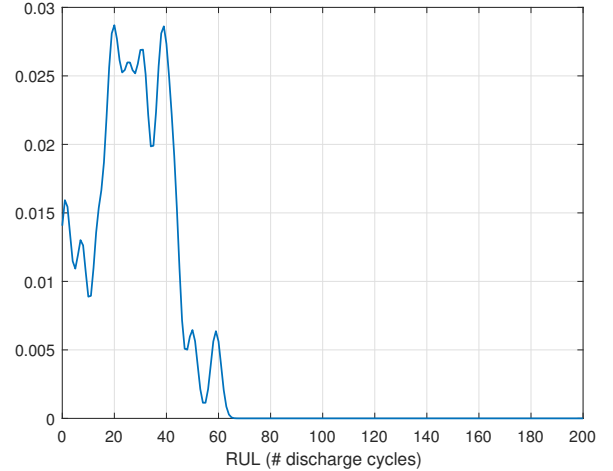


Fig. 5. Predicted probability distribution of the RUL at the end of 80th charge-discharge cycle. The true number of remaining cycles is 29.

Finally, we evaluate a hard estimate of the number of remaining charge-discharge cycles of the test battery at the end of each discharge cycle by taking the mean of the corresponding RUL distribution. The absolute estimation error is shown in Fig. 6. As can be observed from the plot, with only few exceptions, the estimation error is below 20 charge-discharge cycles even at the beginning of battery's life. In fact, the time span from the beginning until the mid of the battery's useful life is essentially where the proposed methodology

provides useful estimates. As the battery gets closer to the end of its useful life, the estimated remaining number of charge-discharge cycles becomes unreliable. We hypothesize that as the battery approaches its end of life the model parameters do not capture all the subtleties in the discharge voltage waveform so as to aid accurate prediction of the remaining number of cycles. Further consideration of this issue and effort to find better empirical model for this regime are left as future research.

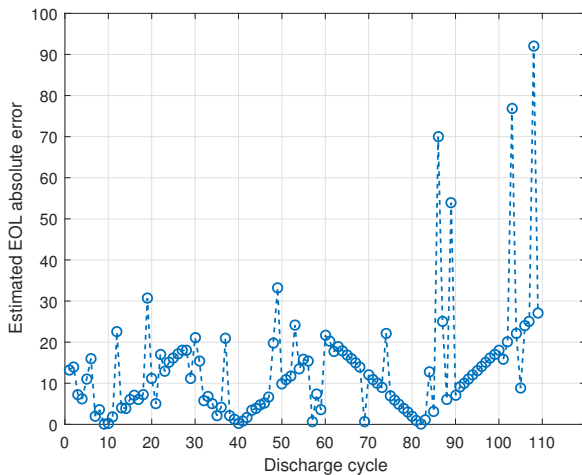


Fig. 6. Absolute estimation error of the remaining number of charge-discharge cycles of test battery B5.

VI. CONCLUSION

This paper presents a methodology for inferring probability distribution of the remaining number of charge-discharge cycles a battery can undergo before it is declared dead. Accurate prediction of the remaining useful life is important in numerous applications since it ensures making timely decision as to when a battery should be replaced such that power interruptions of the system it supplies are avoided. The methodology is based on Dirichlet process mixture model, which is used to cluster feature vectors that represent discharge voltage waveforms of one or more batteries of similar type measured over their life spans. The inference of the DPMM generative model is performed using variational Bayes, which is faster and provides better means to check for convergence than sampling-based inference methods. In the operational stage, the discharge voltage of a battery is measured and fitted using the selected empirical model. The obtained feature vector is then classified into one of the DPMM clusters and the probability distribution of the remaining useful life, expressed as the number of remaining charge-discharge cycles, is inferred. The developed method is tested with experiments, which show that it provides useful predictions of the remaining number of charge-discharge cycles of a battery relatively early in its life.

REFERENCES

- [1] G. O. Sahinoglu, M. Pajovic, Z. Sahinoglu, Y. Wang, P. Orlik and T. Wada, "Battery State-of-Charge Estimation Based on Regular/Recurrent Gaussian Process Regression", *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, May 2018.
- [2] M. Pajovic, Z. Sahinoglu, Y. Wang, P. Orlik and T. Wada, "Online data-driven battery voltage prediction", *IEEE 15th International Conference on Industrial Informatics (INDIN)*, July 2017.
- [3] L. Wu, X. Fu and Y. Guan, "Review of Remaining Useful Life Prognostics of Vehicle Lithium-Ion Batteries Using Data-Driven Methodologies", *MPDI Applied Sciences*, 2016.
- [4] B. Saha, K. Goebel, S. Poll and J. Christophersen, "Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework", *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 2, Feb. 2009.
- [5] Z. Liu, G. Sun, S. Bu, J. Han, X. Tang, M. Pecht, "Particle Learning Framework for Estimating the Remaining Useful Life of Lithium-Ion Batteries", *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 2, Feb. 2017.
- [6] D. Wong, F. Yang, K. L. Tsui, Q. Zhou and S. J. Bae, "Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Spherical Cubature Particle Filter", *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 6, June 2016.
- [7] M. E. Orchard, L. Tang, B. Saha, K. Goebel, G. Vachtsevanos, "Risk-Sensitive Particle-Filtering-based Prognosis Framework for Estimation of Remaining Useful Life in Energy Storage Devices", *Studies in Information and Control*, vol. 19, no. 3, Sept. 2010.
- [8] Q. Miao, L. Xie, H. Cui, W. Liang and M. Pecht, "Remaining useful life prediction of lithium-ion battery with unscented particle filter technique", *Microelectronics Reliability* 53 (2013) 805–810.
- [9] B. E. Olivares, M. A. Cerda Munoz, M. E. Orchard and J. F. Silva, "Particle-Filtering-Based Prognosis Framework for Energy Storage Devices With a Statistical Characterization of State-of-Health Regeneration Phenomena", *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 2, Feb. 2013.
- [10] M. E. Orchard and G. J. Vachtsevanos, "A particle-filtering approach for on-line fault diagnosis and failure prognosis", *Transactions of the Institute of Measurement and Control* 31, 3/4 (2009) pp. 221-246.
- [11] W. Yan, B. Zhang, G. Zhao, J. Weddington and G. Niu, "Uncertainty Management in Lebesgue-Sampling-Based Diagnosis and Prognosis for Lithium-Ion Battery", *IEEE Transactions on Industrial Electronics*, vol. 54, no. 10, Oct. 2017.
- [12] Y. Jinsong, L. Shuang, T. Diyin and L. Hao, "Remaining Discharge Time Prognostics of Lithium-Ion Batteries Using Dirichlet Process Mixture Model and Particle Filtering Method", *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 9, Sept. 2017.
- [13] J. Joseph, F. Doshi-Velez and N. Roy, "A Bayesian Nonparametric Approach to Modeling Battery Health", *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012.
- [14] D. M. Blei and M. I. Jordan, "Variational Inference for Dirichlet Process Mixtures", *Bayesian Analysis*, 2006.
- [15] Y. W. Teh, M. I. Jordan, M. J. Beal and D. M. Blei, "Hierarchical Dirichlet Processes", *Journal of the American Statistical Association*, 101(476):1566–1581, Nov. 2005.
- [16] Y. W. Teh, "Dirichlet Processes", *Encyclopedia of Machine Learning*, Springer, 2010.
- [17] C. E. Antoniak, "Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems", *The Annals of Statistics*, vol. 2, pp. 1152-1174, 1974.
- [18] B. Saha and K. Goebel, "Battery Data Set", *NASA Ames Prognostics Data Repository*, NASA Ames, Moffett Field, CA, 2007.